Title	関数計算について (数値計算のアルゴリズムの研究)
Author(s)	一松, 信
Citation	数理解析研究所講究録 (1972), 149: 1-13
Issue Date	1972-06
URL	http://hdl.handle.net/2433/106782
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
149巻 1972年 1-13

関数計算について

京都大学・数理解析研究所
一松　信

I.　連分数について

関数計算の概論については，文献[1]，[2]，[3] などに詳しい．ここでは主として有理関数近似，とくに連分数の利用について解説する．別項東大大型機センターでの原案作成の報告にもあるように，平方根以外の初等関数の原案としては，連分数の利用がもっともすぐれているし，各種特殊関数の計算にも，大いに有用である．なお関数項の連分数に関する参考書として，[4]，[5]，[6] がある

1.1. 連分数

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots}}}$$

の形の式を連分数という．項 a_n, b_n は数でも多項式でもよい．このままでは場所をとるので，いろいろ略記法が使われ

る．たとえば

$$b_0+\frac{a_1}{b_1+\frac{a_2}{b_2+\frac{a_3}{b_3+\cdots}}}$$

$$b_0+\frac{a_1}{b_1}\dot{+}\frac{a_2}{b_2}\dot{+}\frac{a_3}{b_3}+\cdots$$

$$b_0+\frac{a_1|}{|b_1}+\frac{a_2|}{|b_2}+\frac{a_3|}{|b_3}+\cdots \quad =b_0+\mathop{\Phi}_{n=1}^{\infty}\frac{a_n|}{|b_n}$$

など．ここでは最後の形を使うことにする．これを $\frac{a_n|}{|b_n}$ の項でうち切った式を，<u>第 n 近似分数</u> という．これを整理して $\frac{P_n}{Q_n}$ と書けば，分母，分子はともに同じ形の<u>漸化式</u>

$$P_n=b_nP_{n-1}+a_nP_{n-2},\quad Q_n=b_nQ_{n-1}+a_nQ_{n-2},$$

$$P_{-1}=1,\ P_0=b_0,\quad Q_{-1}=0,\quad Q_0=1$$

をみたす．$b_0=0$ のときには，

$$P_0=0,\ P_1=a_1,\quad Q_0=1,\ Q_1=b_1$$

からはじめることが多い．（これは n に関する帰納法で，すぐに証明できる．） これを利用して，連分数をいろいろ変形できる．もし項数 n が既知なら，n の大きい方から

$$F_n=b_n,\quad F_k=b_k+\frac{a_k}{F_{k+1}}\quad (k=n-1,\cdots,2,1,0),$$

$$F_0=\frac{P_n}{Q_n}$$ という反復も可能だが，除法を多く要する欠点がある．

1.2. S 連分数

関数計算によく使われるのは，S連分数（Stieltjes の連分数）とよばれるもので，

$$\frac{c_0|}{|1} + \mathop{\Phi}_{n=1}^{\infty} \frac{c_n x|}{|1}$$

の形のものである．

$f(x) = \sum_{n=0}^{\infty} \alpha_n x^n$ （Taylor 展開）が既知なら，係数 c_n は，これから下記の商差法で計算できる．順次

$$q_n^{(1)} = \frac{\alpha_{n+1}}{\alpha_n}, \qquad e_n^{(0)} = 0$$

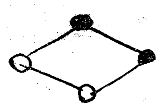
$$e_n^{(k)} = \left(q_{n+1}^{(k)} - q_n^{(k)}\right) + e_{n+1}^{(k-1)}, \qquad q_n^{(k+1)} = \frac{e_{n+1}^{(k)}}{e_n^{(k)}} q_{n+1}^{(k)}$$

とおくとき，

$$c_0 = \alpha_0, \quad c_{2k-1} = -q_0^{(k)}, \quad c_{2k} = -e_0^{(k)}$$
$$(k = 1, 2, \cdots)$$

この計算は，下記のような商差表によるとよい．

$$\begin{array}{lllll}
\alpha_0 \to c_0 & & & & \\
 & q_0^{(1)} \to -c_1 & & & \\
\alpha_1 & & e_0^{(1)} \to -c_2 & & \\
 & q_1^{(1)} & & q_0^{(2)} \to -c_3 & \\
\alpha_2 & & e_1^{(1)} & & e_0^{(2)} \to -c_4 \\
 & q_2^{(1)} & & q_1^{(2)} & \\
\alpha_3 & & e_2^{(1)} & \vdots & \vdots \\
 & q_3^{(1)} & \vdots & & \\
\alpha_4 & \vdots & & & \\
\vdots & 商 & 差 & 商 & 差
\end{array}$$



ひし形の右上2個と左下2個の積または和が等しいように作る．

実用上では，桁落ちを防ぐ工夫のいることが多い．

この計算は，一種の互除法であって，S連分数の第n近似分数を $x=0$ でTaylor展開したものが，なるべく高次の項まで $\sum \alpha_n x^n$ とあうように順次作ったものである．ただし $e_n^{(k)} \neq 0$ で，計算が続行できることは仮定する．

$f(x)$ の Padé近似，すなわち p, q を定めて

$$\left(\sum_{i=0}^{q} b_i x^i\right)\left(\sum_{n=0}^{\infty} \alpha_n x^n\right) - \sum_{j=0}^{p} a_j x^j$$

が，なるべく高次の項まで消えるように a_j, b_i を定めると，$p=n-1, q=n$；$p=n, q=n$ に対する Padé近似

$$\sum_{j=0}^{p} a_j x^j \Big/ \sum_{i=0}^{q} b_i x^i$$

が，ちょうど，$f(x)$ のS連分数の第 $2n-1$；$2n$ 近似分数に相等しい．したがって，これからも，$f(x)$ のS連分数がもとめられる．$\sum \alpha_n x^n$ は必ずしも収束するTaylor展開でなくても，漸近展開でもよい．

初等関数（および古典的特殊関数）の各種の連分数展開は，上記の方法以外に，漸化式の反復や，ある種の Riccati 型微分方程式を反復して解く手法から導びかれる．最後の方式は Lagrange に負うもので，[6] に詳しく記述されている．

1－3．実例

第1種 Bessel 関数 $J_\nu(x)$ について，

◎ $$\frac{J_\nu(x)}{J_{\nu-1}(x)} = \frac{x|}{|2\nu} + \underset{n=1}{\overset{\infty}{\Phi}} \frac{(-x^2)|}{|2(\nu+n)}$$

これは漸化式の反復で導びかれる. とくに $\nu=1/2$ とすると

$$\tan x = \frac{x|}{|1} + \underset{n=1}{\overset{\infty}{\Phi}} \frac{(-x^2)|}{|2n+1},$$ x を ix におきかえて

$$\tanh x = \frac{x|}{|1} + \underset{n=1}{\overset{\infty}{\Phi}} \frac{x^2|}{|2n+1}$$

これらは三角関数, 指数関数の原器用として有用である.

◎ $$\int_0^x \frac{dt}{1+t^k} = \frac{x|}{|1} + \frac{x^k|}{|k+1} + \underset{n=1}{\overset{\infty}{\Phi}} \left[\frac{n^2k^2x^k|}{|2nk+1} + \frac{(nk+1)^2x^k|}{|(2n+1)k+1} \right]$$

とくに $k=1, 2$ の場合を変形整理すると,

$$\log(1+x) = \frac{x|}{|1} + \underset{n=1}{\overset{\infty}{\Phi}} \left[\frac{nx|}{|2} + \frac{nx|}{|2n+1} \right]$$

$$\arctan x = \frac{x|}{|1} + \underset{n=1}{\overset{\infty}{\Phi}} \frac{n^2x^2|}{|2n+1}.$$

◎ $$x^{-\nu} e^x \int_x^\infty t^{\nu-1} e^{-t} dt = \frac{1|}{|x} + \underset{n=1}{\overset{\infty}{\Phi}} \left[\frac{n-\nu|}{|1} + \frac{n}{x} \right].$$

とくに $\nu=0$ として整理すると

$$\int_x^\infty \frac{e^{-t}}{t} dt = e^{-x} \left[\frac{1|}{|x+1} - \underset{n=1}{\overset{\infty}{\Phi}} \frac{n^2|}{|x+2n+1} \right]$$

$\nu=1/2$ として変形すると

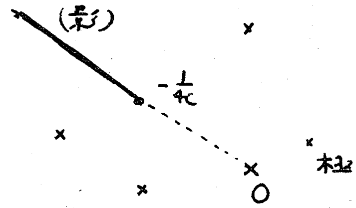
$$\int_x^\infty e^{-t^2/2} dt = e^{-x^2/2} \left[\frac{1|}{|x} + \underset{n=1}{\overset{\infty}{\Phi}} \frac{n|}{|x} \right]$$

これらは積分指数関数や誤差関数の中間の x に対する計算に有用である.

1-4. S連分数の収束

これには非常に多くの定理がある（とくに[5]に詳しい）.
S連分数について、重要な結果を証明なしにのべる.

$0 \leqq c_n \leqq g$ ならば、S連分数（の部分分数の列）は $|x| < 1/4g$ で収束する.



$c_n \to c$ $(n \to \infty)$ ならば、
S連分数は、複素数平面から、
0, $-1/4c$ を結ぶ半直線上の $-1/4c$
を越える影の部分を除いて収束する. さらに（この領域内で）$f(x)$ の極を含まないコンパクト集合上で、一様に収束する. とくに $c=0$ ならば、影を空とみなす. また $c_n \geqq 0$, $c=+\infty$ ならば、影を負の実軸と解釈して正しい.
（Van Vleck の定理. [5]参照）.

漸化式から

$$\frac{P_n}{Q_n} - \frac{P_{n+1}}{Q_{n+1}} = \frac{(-1)^{n+1} a_1 a_2 \cdots a_{n+1}}{Q_n Q_{n+1}}$$

であるから、Q_n のおよその評価があれば、打切り誤差が評価できる. とくに $a_n > 0$, $b_n > 0$（S連分数なら $c_n > 0$ で $x > 0$）のときには、部分近似分数は、交互に過大、過小の近似値をとり、振動しながら収束するから、誤差の評価は容易である.

連分数展開は極の向う側まで有効なのに注意する. たとえば, $\tan x$ の展開は, 極 $\frac{\pi}{2}$ をこえて, 1.6 とか 2.0 でも使える. ——じっさいには $\frac{\pi}{4}$ までで, あとは換算すべきであるが.

前記の Taylor 級数 $\sum \alpha_n x^n$ は, たとえ収束半径が 0 であっても, あまりに急激に $|\alpha_n|$ が大きくなるのでなければ, ——たとえば $\sum_{n=1}^{\infty} |\alpha_n|^{-1/2n} = \infty$ という程度であれば, 書きかえた連分数は空でない収束域をもつ. $|\alpha_n| \leqq (n!)^{\lambda}$, $\lambda \leqq 2$ ならば, この条件が成立する. 合流型超幾何関数の一族 (Bessel 関数, 不完全ガンマ関数など) の, ∞ での漸近展開を $\frac{1}{x}$ の S 連分数に書きかえたものは, この性質をみたす. したがってこれは一種の総和法ともみなされ, 関数値計算にも有用である. ただし, この種の連分数の収束は一般に遅い. 不完全ガンマ関数など, c_n が n の 1 次式程度に増大するものでは, n 項までの打切り誤差の主要項は, ほぼ $C\varepsilon^{\sqrt{n}}$ の形になる ([7]). したがって, 先へゆくほど項数のふえ方がますので, あまり高精度を望むのは不適当であり, 適当な x と精度の範囲で使うべきである.

II　関数値計算に関するメモ

2.1 近似式の選択

必ずしも最良近似にこだわらず，むしろ展開式などのほうが，数値の誤り，誤植，うつし誤り，ミスパンチなどの危険をさけるために安全である．区間をとくに短くすれば，展開式も最良近似式も大差ないし，単長，倍長，4倍長などと精度の変更も容易である．この目的の為めに，係数値などを十分検査したデータカードの形で販売しては，という提案もある（需要の点はともかくとして）．

2.2 関数の定義限界

末尾のビットまで正しく計算したいのはやまやまであるが，引き数の精度上，それは無意味なことがある．|X|の大きな引き数に対しては，SIN(X)，COS(X)の値は無意味であるし，EXP(X)の末位の桁も無意味に近い．またべき乗は注意を要する．複素数の計算で (−1.0)**(1.0/3.0) が −1 にならず，複素数 $\frac{1}{2}+\frac{\sqrt{3}}{2}i$ となるのは，定義からいえばむしろ当然である．

16進法の機械では，末位の桁は信用できないものと覚悟したほうがよい。とくに2進で正規化されない定数 π/2，

0.1 などには注意を要する。なるべく引き数，値とも精度が保たれるようなスケーリングが必要なことがある．

2.3. 計算法上の注意

近似式そのものには十分精度があっても，それを計算する途中で精度がおちては何にもならない．極度に大きな係数の現われる近似式や，符号が交互の多項式近似式は警戒を要する．0に近い関数値は，(0に近い量)×(他の量)の形で計算すべきであって，断じて相近い2量の加減で計算すべきではない．$x_0 \neq 0$ で $f(x_0)=0$ となる関数の x_0 の近くは，特別の工夫がいる．$\log x$ の $x=1$，$\cos x$ の $x=\frac{\pi}{2}$ などがその好例で，下手にやると大きな相対誤差がまぎれこむ．

また多項式の計算を，ふつうにやる Horner の反復法でなく、x^2 でまとめて演算（とくに乗算）回数をへらす工夫がいろいろ研究されているが，じっさいに使った例では，結果の精度や安定性などの点で感心しないことが多いとのことである．サブルーチンも常用の早いもののほか，時間はかかっても完全に末位まで正確に，という原器用のものも必要であろう．

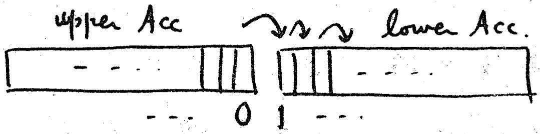
III. FORTRANで金物のミスを発見した話

1971年10月に数理解析研究所付属，数理応用プログラミング施設長をおうせつかり，計算室の（名儀上の）責任者になってすぐにおこったこの話をのべたい．この中には関数計算に対する教訓も含まれているから．・・・

ことのおこりは，季節のかわり目に，朝スイッチを入れてから安定するまでに，しばしば演算ミスが発生することにはじまる．皮肉なことに，メーカーから提供されたテストプログラムはうまく通る．ひっかかったのは占部教授が作られたもので，いろいろの範囲の数（2進法で有限小数になるもの）xに対する$\sqrt{x}$を作り，$(\sqrt{x})^2-x$を作って，その上限をおさえるものである．これが正しい値におちつくまで，しばしばその上限が変動し，しかも正しい値の10^5倍以上という桁違いの誤りを生ずる．

はじめ2乗をY**2にしたのがいけないのかと思ってY*Yに直したが，まったく差がない．そのうちに$\sqrt{x}$の値自体がおかしいことに気がついた．皮肉なもので，同じテストプログラムを何回もかけると，なかなかよくならない．少しおかしいがやってみろ，と他のプログラムをかけて「気分転換」(?)させると，一度によくなったりした．この分では気嫌直しの特効薬(?)ができそうだ，と冗談もいったりした．

そのうちに $\sqrt{x}$ の値がおかしいのは，引き数 x のパタンがある数種類の型に限ることがわかった（8進のダンプで）．そこでその値を入れ，正常に動作するときの正解を作っておいて，トレーサーで追跡してみた．しかしここでわれわれは肩すかし（?）にあった．大捕物陣（?）をしいた前夜，磁気テープ装置の故障を夜半すぎまでかかって修理したので，CPUの睡眠不足（?）のため，その日は全然誤動作が生じなかった！やむなく休日の翌日，十分「休養」させてから再度ためしてみた．その結果，誤りは除法で生ずること，しかも1語24ビット，2語で浮動小数点数を表わしているが，lower Acc.の最下ビットが1，その上が0のとき，最後に剰余の正規化をしてシフトすると，lower Acc.の最上部が01になるべきなのが，ときたま11となる誤動作であることが確認された．



$\sqrt{x}$ のサブルーチンは，近似式で初期値を定め，Newton法の反復を3回くりかえすように作られていた．第1回目に誤りが生じても，あとには影響しない．第2回目の誤りは，最後の答の下位に影響するが，十進法に変換すると，数値上の差は目に見えない．第3回目に生じた誤は致命的である．もしかすると，第1回目に誤りが生じたのが，見すごされてい

た可能性もあった．またこのプログラムでは，2で割るのに0.5をかけたら，指数部を1へらす，といった器用なまねはせずに，正直に定数2.0で割っていた．そしてたった1回だけ，2.0で割る除法で誤りが発生した．

ここまでは専らFORTRANと8進法のダンプで追いつめたものであったが，このあとはもはやソフト屋の手を離れる．あとは調整員が除法を1ステップずつ追っては部品をしらべ，ついに劣化したトランジスタを発見して交換してくれた．それ以後この種のエラーは，まったく発生しなくなった．

今回はまことに好運であった．その上計算室の操作員も，メーカーの調整員も，きわめて有能かつ良心的で，よく働いてくれた．発見以来2週間ほどで原因をつきとめて，完全に修理できたのだから，成功といってよいだろう．私にもよい勉強になった．FORTRANというかなり高級言語だけでも，金物段階の動作がわかるという経験をつんだことと，精度が必要なら，収束するまで反復をくりかえすほうが安全なことの確認である．もしも収束するまで反復するNewton法で$\sqrt{x}$のプログラムが作られていたら，この発見は困難だったであろう．もちろん金物の誤動作まで考慮にいれてプログラムを作ることは，ソフト屋の守備範囲の逸脱であるが．——

参考文献

[1] 山内・宇野・一松編, 電子計算機のための数値計算法 Ⅲ, 培風館, 近刊.

[2] 一松 信, 数値解析, 税務経理協会, 1971.

[3] D.C.Handscomb編, Methods of Numerical Approximation, Pergamon Press, 1966.

関数項連分数の参考書としては,

[4] O.Perron, Die Lehre von den Kettenbrüchen, 1929, Chelsea, 1966.

[5] H.S.Wall, Analytic theory of continued fractions, 1949, Chelsea, 1967.

[6] A.N.Khovanskii (P.Wynn 英訳), The application of continued fractions and their generalizations to problems in approximation theory, Noordoff, 1963.

商差法と打切り誤差については, たとえば,

[7] P.Henrici - P.Pfluger, Truncation error estimate for Stieltjes fractions, Numerische Math., 9 (1967), 120-138.